Bolt Beranek and Newman Inc.

ⴱⴱⴖ

(12)
b.s.

Research in Natural Language Understanding

Quarterly Progress Report No. 2, 1 December 1977 to 28 February 1978

April 1978

D D C

RECEIVED

MAY 15 1978

B

AD A053958

AD No.
DDC FILE COPY

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN ~~Report No.~~ 3797	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) RESEARCH IN NATURAL LANGUAGE UNDERSTANDING Quarterly Technical Progress Report No. 2, 1 December 1977 – 28 February 1978		5. TYPE OF REPORT & PERIOD COVERED Quarterly Technical Progress Report
		6. PERFORMING ORG. REPORT NUMBER Report No. 3797
7. AUTHOR(s) W. A. Woods William A. Woods		8. CONTRACT OR GRANT NUMBER(s) N00014-77-C-0378 ARPA Order-3414
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 50 Moulton Street Cambridge, MA 02138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 7D30
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA 22217		12. REPORT DATE APR 78
		13. NUMBER OF PAGES 30 33 p.
14. MONITORING AGENCY NAME & ADDRESS(If different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, If different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Distributed activation memory networks, knowledge representation, marker passing algorithms, natural language understanding, parsing, semantic interpretation, semantics, structured inheritance networks, taxonomic structures.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is the second quarterly progress report of the ARPA-sponsored Natural Language Understanding project at BBN. The goals of the project are to develop techniques required for fluent and effective communication between a decision maker and an intelligent computerized display system in the context of complex decision tasks such as military command and control. This problem is approached as a natural language understanding problem, since most of the techniques required (cont'd. over)

DD FORM 1473   EDITION OF 1 NOV 65 IS OBSOLETE
1 JAN 73

Abstract (cont'd.)

would still be necessary for an artificial language designed specifically for the task. Characteristics that are considered important for such communication are the ability for the user to omit detail that can be inferred by the system and to express requests in a form that "comes naturally" without extensive forethought or problem solving. These characteristics lead to the necessity for a language structure that mirrors the user's conceptual model of the task and the equivalents of anaphoric reference, ellipsis, and context-dependent interpretation of requests. These in turn lead to requirements for handling large data bases of general world knowledge to support the necessary inferences. The project is seeking to develop techniques for representing and using real world knowledge in this context, and for combining it efficiently with syntactic and semantic knowledge. This report discusses three questions involved in representing and using natural conceptual information:

(1) What kind of thing is a concept?

(2) How can semantic interpretation information be represented appropriately in an associative network memory model?

(3) What kinds of marker-passing algorithms can be used to perform incremental semantic interpretation concurrently with parsing in a distributed activation memory network containing large numbers of interpretation rules?

ACCESSION for

NTIS   White Section
DDC   Buff Section ☐
UNANNOUNCED ☐
JUSTIFICATION

BY
DISTRIBUTION/AVAILABILITY CODES
Dist.   AVAIL and/or SPECIAL

A

RESEARCH IN NATURAL LANGUAGE UNDERSTANDING

Quarterly Technical Progress Report No. 2

1 December 1977 to 28 February 1978

ARPA Order No. 3414

Program Code No. 8D30

Name of Contractor:
  Bolt Beranek and Newman Inc.

Effective Date of Contract:
  1 September 1977

Amount of Contract:
  $301,377

Contract No. N00014-77-C-0378

Contract Expiration Date:
  31 August 1978

Short Title of Work:
  Natural Language Understanding

Principal Investigator:
  Dr. William A. Woods
  (617) 491-1850 x361

Scientific Officer:
  Gordon D. Goldstein

Sponsored by
Advanced Research Projects Agency
ARPA Order No. 3414

TABLE OF CONTENTS

Introduction

In our previous report, we set the context for the research to be conducted under the BBN ARPA project in Natural Language Understanding and outlined its goals. Specifically, we are concerned with discovering and developing techniques for dealing with large bodies of information in the kinds of complex decision-making situations that arise in military command and control. Central to this effort are representational conventions for storing natural conceptual information in a machine and algorithms for efficiently using that information. In this report we discuss three fundamental questions:

(1) What kind of thing is a concept?

(2) How can semantic interpretation information be represented appropriately in an associative network memory model?

(3) What kinds of marker-passing algorithms can be used to perform incremental semantic interpretation concurrently with parsing in a distributed activation memory network containing large numbers of interpretation rules?

Towards a Notion of Concept

W.A. Woods

1. The Notion of a Description

In searching for a relatively concrete, understandable framework in which to build a theory of concepts, I have finally settled on the notion of **description** as that which is least likely to be misinterpreted as to its intended meaning. In this framework, the **function** of a concept is to serve as a description of something. This functional sense of description is to be contrasted with the notational and structural features of its **representation** as a string of symbols or assembly of nodes and pointers. I will think of a concept as functioning as a description: it will not necessarily be a description of a thing or of a set, but merely of a situation or some aspect of a situation. I am looking for a notion that will not presuppose models of entities, physical objects, sets, propositions, individual constants, abstractions or any of the usual foundations on which logics are constructed.

Since psychological experiments indicate that sufficiently young children do not have the same model of physical objects and object constancy that adults have and that apparently such models have to be learned, I am looking for an account of concepts and conceptualization in which such models can be learned rather than presupposed. What I will take as a foundation is that the

conceptual system is used by some organism (or artificial system) to classify its experiences, build up its model of the world, and then use that model as a basis for its choice of actions. Given this assumption, the notion of an experienced situation is a reasonably concrete point at which to begin. I will take as a central operation the process of determining whether a given concept (i.e., a description) is satisfied within a situation or within some aspects of a situation. The fundamental relationship then will be the **satisfaction of a concept in a situation.** (I have chosen the wording carefully here so that satisfaction "in" a situation does not presuppose that the situation is an entity which does the satisfying.)

By associating the word "concept" with the notion of a description to be satisfied in a situation, I specifically mean to avoid interpreting a concept as either a set of things or a predicate that is true of them. It will be **possible** to form concepts for sets of things and for predicates, but a concept in and of itself is merely a description. Associated with each concept, there is clearly a second concept - the set of things that satisfy it - and a third concept - the predicate that is true of all the members of that set - but these are both different from the original concept itself. Several of the difficulties that I am seeking to avoid are illustrated by the following example.

Consider the now classical definition of an "arch" as a configuration of three bricks, one of which is supported horizontally on top of two vertical ones above an open space between them [Winston, 1970].　Looking at this concept as a description to be satisfied in a situation, we might use the following notation to write that description:

```
x,y,z : (AND (brick x)(brick y)(brick z)
             (horizontal x)(vertical y)(vertical z)
             (support y x)(support z x)
             (NOT (in-contact y z)))
```

where x, y, and z are particular aspects of that situation.

There are two distinctly different predicates that we can make from this description, both of which characterize arches -- one is a predicate which is true of three bricks, and the other is a predicate that is true of an arch. That is, the former merely characterizes a relationship among three bricks, but does not necessarily thereby postulate the existence of a "thing" called an arch. The second applies to a thing that has already been conceptualized as an object and characterizes whether it is an arch or not. If we call the former predicate ARCH-CONFIG, then the above concept of arch could be abbreviated:

```
x,y,z : (ARCH-CONFIG x y z)
```

and the second predicate could be defined (using Church's lambda operator) as:

```
(LAMBDA (x) (E y)(E z)(E w)(AND (part-of y x)
                               (part-of z x)
                               (part-of w x)
                               (ARCH-CONFIG y z w)
                               (A v)(OR (NOT (part-of v x))
                                        (subpart v y)
                                        (subpart v z)
                                        (subpart v w))))
```

Notice, that when one attempts to spell out this latter concept
exactly - that is, the second arch predicate - a number of
subtleties must be made explicit. The first is that the
individual bricks have to be identified as parts of the object,
but more significantly we have to rule out the possibility of
there being other parts which change the object into something
else (e.g., an arcade or a ladder). The restriction that all
other parts of the arch must be subparts of the three bricks
allows for the bricks themselves to be composed of parts (and
even to have quite elaborate substructure), while assuring that
the object x must not have any other significant parts outside of
y, z, and w. (We assume that the subpart relation used here is
inclusive, so that objects are considered subparts of
themselves.)

Notice that one can perceive instances of arches in things
like ladders and arcades, that the concept of arch is satisfied
in these situations, and that the first arch predicate holds for
various argument assignments in such situations, but the second
arch predicate is not satisfied except by an object previously
conceptualized as a single entity that is itself an arch.

- 5 -

The above example should now have brought the notion of an "object" sufficiently into focus for the following point to be understandable. What makes something an object has nothing to do with the reality of the world or of the sensory experience, but is the result of the way such an experience is conceptualized. I will not be concerned with any philosophical considerations of whether objects really exist in the world outside of our conceptualization. In the theory that I am developing, which I believe will account for our ordinary language notion of "object", objects exist by virtue of an act of conceptualization that may or may not be taken.

2. The Conception of an Object

I will assume a lattice of concepts organized into an inheritance structure (SI-Net) similar to that described in [Woods and Brachman, 1978]. In this structure, each concept is explicitly linked to all the more general concepts that subsume it, with all the relationships among the various **dattrs** (generalized attributes and parts) of those concepts indicated explicitly. Each such concept will be interpreted as a description that may be satisfied in a situation. I will assume that in an experienced situation, the individual aspects of the situation are somehow labeled so that they can be grouped together to form objects if the perceptual system chooses to so conceptualize them. For example, such aspects may be labeled by their position in the visual field when looking in a given

direction at a given time. The grouping of such aspects into
objects will be a mental activity performed by the organism
rather than something objectively "real". This activity is not
un'ike the grouping of the words in a sentence into noun phrases
and verb phrases. Structure is imposed on a sentence by the
grammar used to parse it, and is not an intrinsic attribute of
the sentence itself.

Representing Interpretations as "Cables"

W. A. Woods

The parallel inheritance of **dattrs** from a more general concept by a more specific one in the network can be thought of as a **cable** - i.e., a ribbon of parallel conductors such as connects computer components together. (This metaphor is due to Ron Brachman and Brian Smith). Quillian's [1968] simple notion of a **Superc** connection from one concept to another is thus replaced by a notion of a cable connection which contains separate and distinct inheritance paths for each of the dattrs that are passed down. Given this view, it is clear that there are many different varieties of cables depending on the number of conductors in the cable and the type of inheritance each conductor transmits. (This analogy breaks down somewhat in the case where dattrs are not modified or differentiated in any way and therefore need not be explicitly represented for the more specific node. Fahlman's [1977] notion of a "virtual copy" is a useful way to think about the inheritance of such dattrs by concepts. That is, processing operations on the network are designed to act **as if** a copy of the more general dattr appeared at the more specific node, although no such copy actually exists.)

The idea of a cable as a parallel bundle of inheritance connections is a useful metaphor. We find it occurring again when we consider the correspondence between certain different but

related concepts - e.g., between the internal representation of a sentence describing an instance of an activity and the internal representation of the activity itself. In exactly the same way that we have use SI-Net conventions to characterize conceptualizations of processes, activities, events, physical objects, etc. [Woods and Brachman, 1978], we can also use them to characterize various sentence or utterance types, abstract propositions, facts, etc. Thus we can develop a conceptual taxonomy of sentence types as well as a taxonomy of the kinds of things such sentences describe and make assertions about. It is the correspondence between these two taxonomies that enables us to understand sentences -- that is to construct a representation of what a sentence means from the representation of the sentence itself. This correspondence represents at least one aspect of the process of semantic interpretation, the mapping of English sentences onto representations of what they mean.

For example, let us suppose that an understanding system has learned a concept for "mutual exchange", in particular for the exchange of goods for a monetary token of some kind (say a clam, a la the B.C. cartoon strip). Further, suppose that this concept has been learned without the benefit of language, say by gestures or whatever. Now suppose that we are attempting to teach this system how to describe an instance of this concept in English. We assume that the system already understands basic English grammar - the syntactic categories of words (nouns, verbs, etc.),

the kinds of syntactic constructions (noun phrases, verb phrases, relative clauses, etc.), the various syntactic roles (subject, object, etc.) that constituents can play in larger constructions, etc.

We will say that a non-negative English sentence whose subject can be interpreted as a person, whose main verb is either "sell" or some inflected form of it, whose object can be interpreted as goods, and whose indirect object (if present) can also be interpreted as a person, can itself be interpreted as an instance of "mutual exchange". (Notice that this is a recursive definition of the interpretation of a sentence in terms of the interpretations of its noun phrases, which will need similar rules of interpretation, down to some "atomic" level of direct interpretation.) In specifying this rule of interpretation, we must indicate the connections between the various constituents of the sentence and the corresponding dattrs of the internal representation of this concept of "mutual exchange". This relationship between the concept of a sentence and the concept that it describes has essentially the same cable structure that we described above for inheritance of properties in a concept lattice. <*1>

- - - - - - - - - - - - - - - - -
<*1>. Being a relationship between two concepts, rather than between what they are concepts of, this is essentially a **meta-description** [Smith, 1978].

This notion of correspondence is not of course an original observation. It appears in one form or another in every language understanding system that constructs some underlying representation from English sentences. Some attempts have even been made to identify this correspondence as an object of study. For example, I think this is what Moore and Newell [1973] try to identify by the term "mapping", although the use of such a general term fails to clearly delineate the idea.

Let me refer to the kind of cable that can be used to connect a representation of a sentence to a representation of its interpretation an **interpretation cable**. Given its function, such a cable can by extension be used to associate other pairs of concepts as well - e.g., to associate a concept of the flag being up on a mailbox with an interpretation that there is mail inside, or the concept of of a dog's mouth watering with an interpretation of it being hungry, or the concept of one person hitting another with an interpretation that the first person is angry at the second. In fact, this cable metaphor can characterize a large class of common inferences that people make.

We might now proceed to ask whether such "cablings" might not themselves be objects capable of being conceptualized. And indeed, we find ample evidence of such conceptualizations in the meanings of such English terms as "relationship", "correspondence", "analogy", "mapping and "inference rule". In fact, we can represent in the usual SI-Net notation the structure

- 11 -

of such a "cabling" concept by having dattrs for the concepts at either end of the cable (i.e., its **source** and **destination**) and quantified **structural descriptions** expressing the correspondence relationships between them. However, before we devolve into infinite regress, let us assume that although such cables can be conceptualized in terms of the epistemological primitives of the SI-Net notation, they may nevertheless be required as primitive capabilities by an understanding and reasoning system. Such a system will use them directly (rather than emulating them by means of some more primitive operations).

If one diagrams these interpretation mappings, they can be viewed as a graphic way of illustrating what is going on in an ATN grammar analysis of a sentence. In the process, a relationship is established between each phrase in the sentence and a corresponding role in the underlying structure -- a process that is mediated by an operation that assigns that phrase to a register and a later operation that uses the contents of that register in constructing a larger structure. One can use similar diagrams to illustrate the relationship between a variable instantiated in the match of a pattern->action rule and its use in the action part. However, it is clear that in the usual implementation of such grammars and rules, these correspondences are not represented explicitly. Rather, they are effected via some process that makes use of . termediate variable bindings/register settings. Nevertheless, I believe a case can

be made both for carrying this explicit representation further
than a mere pedagogical device and for implementing it as a
connection in a network representation.

It has been observed by a number of people including
Winograd [1975] that the representation of knowledge by arbitrary
procedures makes it difficult for other procedures to make use of
that knowledge for other purposes. I think that this contention
is somewhat ill-posed, since it is not that one representation
has anything less to do with procedures than another, but rather
that a notation designed for efficient use by one procedural
interpreter does not necessarily facilitate efficient use by
other processes for other purposes. That is, if the only
accessing operation that one process performs on a data structure
is "go to the next step", then a representation chosen for
implementing this data structure with only that process in mind
will not likely provide all the access paths that might be
desired by other processes.

In the case of ATN's, specifying structure building
operations solely for the purpose of building such structures
does not usually provide an reasonable way to ask of a structure
whether it could have been built by some particular portion of
the grammar or what its realization would be in English. In a
similar way, specifying a pattern->action rule to facilitate its
application in the direction of the arrow does not necessarily
help to characterize what structure may result from its

application or how to determine for a given situation whether it could be the result of one or a combination of such rules. This shows up for example in transformational grammars, which are essentially defined by a set of pattern->action rules. In such grammars, the only characterization of the possible legitimate surface structures of sentences is that which is implicit in the set of transformational rules - i.e., those structures that can be built by the rules of the grammar operating on deep structure trees characterized by the base component grammar.

The kind of representation that we have presented above - a representation in which both the input and output ends of a cable are fully specified concepts with the dattr correspondences between them explicitly indicated - allows us to represent pattern->action rules in a way that allows one to follow the correspondences both forward and backward. As such, it may provide the elusive representation in which the essential nature of what is being represented is not being buried or obscured in non-essential detail for some procedural interpreter.

Marker Passing Algorithms

W. A. Woods

Given a **knowledge management system** such as that needed in a
command and control environment, its knowledge base will be used
for different purposes by different components at different
stages in its processing of an input utterance.   For example,
this knowledge base will be used by the parser to choose among
alternative possible parsings and interpretations of an input
utterance;   it will be used to record the user's standing orders
which are to be carried out whenever the system enters a
particular configuration or whenever a particular event occurs;
it will be used to answer both the user's explicit questions as
well as internally posed questions which may be answerable
without resorting to the user; and it will be used by the problem
solver to plan solutions and search for proofs.

In order to implement many of these basic operations, we
will employ a kind of "spreading activation technique" [Quillian,
1968]. Specifically, we will make use of a hypothetical parallel
machine structure with a certain amount of processing capability
at each of its nodes, whose knowledge base is in the form of a
network and which has the ability to pass markers from node to
node and to broadcast instructions to them.   The abstract
architecture that we envisage for this system is similar to that
of [Fahlman, 1977], although we hypothesize a more powerful node
than he does, as well as a greater ability for parallel,

- 15 -

asynchronous activity without the detailed sequential supervision of an overlord controller. We will use the capabilities of such an overlord, however, to regulate and modify the activities of the nodes and to make the ultimate decisions on courses of action.

We will also make the following assumptions: (1) The system will contain a finite number of **marks** (probably numbering a thousand or more) that can be associated with the nodes of the network in a number of different ways. Specifically, each node will have several different registers or **slots** that can hold a given mark, each slot corresponding to a different **status** of the node with respect to the mark. (2) A slot may hold an ordered pair of marks, and moreover a slot can hold several marks (or pairs of marks) simultaneously. (3) Marks can be passed along the links of the network, either singly or in pairs, together with one of a finite number of "subscripts" (probably numbering a dozen or so) indicating how that mark (or pair of marks) is to be processed at the destination node. (4) A node can request a mark from a list of available ones. <*2>   If that mark is put in a slot or status called OWN, then it effectively serves as a temporary handle on that node.   For example, the node could respond to broadcast requests for the owner of that mark.

- - - - - - - - - - - - - - - - - -
<*2>. This may require freeing up and recycling an old mark.

Coupled with the ability to respond to broadcast requests, this use of marks provides a way of implementing a selective addressing scheme. This obviates the need for a node to have an address decodable as a pointer in order to access it. For large semantic networks, fewer bits will be required for such marks than would be required for equivalent pointers, and this in turn will reduce the number of bits of communication required among nodes for various kinds of algorithms.

This use of marks as node handles can be thought of as an alternative way to hold onto intermediate results, as opposed to using temporary registers or variables. In this case, marks are attached to the values themselves rather than registers holding pointers to those values. Not only can this reduce the bit rate of communication, it can potentially remove some contention from communication channels. This is because instructions to mark nodes can be broadcast without requiring pointers to the marked nodes be returned to fixed registers in a central processor. It also has the advantage of permitting the equivalent of multiple (non-deterministic) setting of registers by letting a given mark be assigned to several nodes. Subsequent broadcast instructions to perform operations on the node(s) so marked will automatically be done in parallel for all possible values of this "non-deterministic variable".

1. Virtual Structure-Building with Marks

Another use of marks is to construct temporary connections between different nodes in the network. For example, we can take the simultaneous presense of a given mark m on node A in slot SOURCE and on node B in slot DEST as a **virtual link** from A to B. In order to assign this link specific properties, one can make node M their locus and then associate M with the link via the same mark m. Such virtual links make it is possible to build up short-term memory representations of potential network structure that may or may not be eventually incorporated into long-term memory. Such short-term "scratch" structures can be used in hypothesizing alternative interpretations of input sentences, planning future actions, making internal inference steps, etc. If nothing is done to make such virtual links permanent, they can be made to disappear when their marks become sufficiently old and are collected.

To   illustrate   more   concretely   the   kinds   of   virtual structures that can be built with marks, suppose that as a possible interpretation of a part of an input utterance, we wanted to represent the proposition that the ship Enterprise is located in Boston. Let us suppose that concep, nodes contain enough internal memory to store n-tuples of marks, with some association (either explicit or implicit) between the positions in the n-tuple and the dattrs of the concept. These n-tuples can be used to store temporary individuators of the concept involved.

- 18 -

The first component of the n-tuple, corresponding to the "whole" dattr that stands for an instance of the concept as a whole [Brachman, 1978], will contain a unique mark to be used as a handle on the temporary individuator being constructed. The remaining marks will be the marks for the concepts that are to be used as the fillers for the respective dattrs. Then, assuming that the concepts [Boston] and [Enterprise] have already been marked as owners of the marks m1 and m2, respectively, we can represent the Enterprise being in Boston by an n-tuple (m3 m2 m1) stored under the concept [a ship being located at a place], where m2 is associated with the [ship] dattr of the concept, and m1 is associated with the [place] dattr. The mark m3 can now be used as a handle on the hypothesized individuator, [Enterprise being located in Boston]. If it is later decided to make this information a permanent part of long term memory, a process can be invoked to create a new individuator node whose superconcept is the [a ship being located at a place] node, and whose [ship] dattr is filled with [Enterprise] and whose [place] dattr is filled with [Boston]. <*3>

- - - - - - - - - - - - - - - - - -

<*3>. A possible way of implementing the n-tuples with correspondences to dattrs would be to use nodes in the network that were exactly like individuator nodes, but without any dattrs filled in. Each generic concept could have a certain number of such nodes, reflecting the number of individuators of that concept that might need to be independently hypothesized at one time. Additional such individuators could be created whenever the number of them was found to be insufficient. When a permanent individuator was to be created, the temporary marks associated with the dattrs of one of these special nodes could then be replaced with actual Val links to the concepts which owned those marks, and a new empty individuator of the generic

2. The Interpretation Algorithm

In this section, I will consider a particular marker passing
algorithm for incrementally constructing the semantic
interpretation of a sentence being parsed. I will assume that
the sentence is parsed by an ATN grammar that has access to a
structured inheritance network containing a taxonomy of sentence
constituent types. I will also assume that interpretation cables
connect nodes in this taxonomy with appropriate nodes in a
taxonomy of internal meaning representations (the **MRL taxonomy**).
An interpretation cable will itself be implemented as a concept
with dattrs for its source and destination, as well as zero or
more dattrs for **imap pairs**. An imap pair will consist of a
structure with two dattrs - source and destination - whose values
are pointers to the dattrs of source and destination nodes that
are to correspond under the interpretation assigned by the cable.
The interpretation cable is thus a meta-description that
"mentions" other concepts and their dattrs [Smith, 1978].

We are concerned with a process in which, as the parser
constructs a gradually elaborated syntactic structure in the
syntax taxonomy, corresponding potential interpretations of those
structures are being constructed in the MRL taxonomy. All of
these are assumed to be virtual structures, as described above.

- - - - - - - - - - - - - - - - - -

concept could be created (either immediately, or later, when
needed) to replace the one that has just been used up. Other
implementations are also possible - e.g., providing for internal
storage of vectors of marks with each concept node.

In order to provide for the sharing of initial syntactic
hypotheses that are only later differentiated, we will use the
following conventions for representing a virtual structure.

When the parser begins a new construction, it calls a
function (GETHANDLE <concept node>) whose argument is the node in
the syntax taxonomy corresponding to the type of construction
about to be parsed. GETHANDLE will cause the node to (1) request
that a mark be assigned to it, (2) store that mark in its ROOT
slot and (3) return to the parser. As the parser assigns
constituents to roles in the current construction, it will call a
function

```
(FILL-ROLE <previous handle(ph)>
           <dattr specification>
           <constituent handle(ch)>)
```

which will cause all nodes marked with the <previous handle> **ph**
that have a dattr satisfying <dattr specification> to "fill" that
dattr with the <constituent handle> ch. This "filling" will be
represented by allocating a new mark, call it **nh**, to serve as a
handle on the resulting virtual concept (i.e., the one with the
additional dattr filled. The pair [nh, ph] will then be stored
under this new concept with status MADE-FROM (indicating that it
is "made from" the concept whose handle is **ph**), and the pair [nh,
ch] will be stored under the selected dattr with status
BY-FILLING, indicating that this new virtual concept results from
filling this dattr with the constituent whose handle is **ch**). The
handle **nh** is then returned to the parser. This method of

recording the virtual concept allows the parser to take any given ph and construct several different concepts from it, each differing in its most recent mark.

When a constituent is popped, its handle is popped as well, as a constituent handle for higher constructions. In general, the parser can be non-deterministic, following several alternative paths in parallel, and forking into multiple continuations at any point. The marking algorithm just described provides for both sharing all marks prior to a fork and keeping distinct those marks assigned subsequently or different paths.

The basic algorithm for interpreting structures while parsing is as follows:

1) As each word in the utterance is considered in turn, it is assigned a unique mark in its OWN slot, which will serve as its handle. At the same time, that mark will be assigned to a POSITION slot in a node in a sequence structure that marks the position of that word in the sentence. <*4> The result of this is that for each position in the input sequence, there is a mark that serves as a handle on both that position and the word that occurs there. This double use of the same mark represents the correspondence between the word position and the word, without creating pointers or additional list structure.

- - - - - - - - - - - - - - - - - - - -

<*4>. Alternatively, the nodes in the sequence structure can already have such marks, and the request of the word for a mark can result in its getting the mark from the next available sequence node: many disciplines are possible for managing such marker assignment.

2) As each word is assigned an OWN mark, it will pass that mark up its Superc chains with a subscript that will result in the mark's being associated with each superconcept node in status SUP. It will also pass that same mark, with subscript INTERP! to each node which it can reach via an interpretation cable. This process can itself be broken down into steps of passing the subscripted mark to each interpretation cable of which this node is the source. The cable in turn passes the subscripted mark to its destination node. The result of this process is that each possible interpretation of an input word receives the same mark as that word, with subscript INTERP!. When a node receives such a subscripted mark, it places the mark in a slot called INTERP. The assignment of such a subscripted mark to the INTERP slot of a node causes it to be passed, in turn, through any Superc links from that node. This causes all more general concepts of which this concept is a restriction to receive the same mark, which they will also record in status INTERP. The result will be that all nodes in the MRL taxonomy that subsume a marked concept at the source end of an interpretation cable will receive the mark passed over that cable.

3) As I mentioned above, when the parser hypothesizes a constituent structure, the corresponding node in the syntax taxonomy requests a mark to be allocated, which it records in status ROOT and returns to the parser.

4) When the parser calls (FILLROLE <ph><dattr spec><ch>), hypothesizing that a particular constituent might fill a dattr in a construction being built, the concept that has mark **ph** either as a ROOT or as the first mark of a MADE-FROM pair will try to fill its specified dattr with the concept that OWNs the mark **ch**. This process of trying to fill a dattr will involve verifying the consistency of both the value restrictions on that dattr and any structural conditions involving it. <*5> What the parser will get back from this is either a new mark corresponding to the resulting new hypothesis or a value NIL indicating no such interpretation is consistent. If the assignment of the constituent concept **ch** to this dattr of the construction **ph** is successful, then the concept for the construction will ask for a new mark, **nh**, which it will pair with the old mark **ph** under the status MADE-FROM and return to the parser. This same mark will also be paired with the constituent mark **ch** in status BY-FILLING under the dattr node being filled. The significance of such an new-old pair in MADE-FROM status under a concept is that the new mark is a handle on a new incremental individuator of the concept, consisting of the old individuator plus the specification of an additional dattr. <*6>

- - - - - - - - - - - - - - - - - - -

<*5>. Such consistency checks will themselves be done by other marker passing algorithms.

<*6>. Again, this is only one of many possible disciplines for storing marks associated with a node. It is being given for illustrative purposes. The selection of exactly what marker passing conventions will ultimately be most efficient will require experimentation.

If this new incremental individuator satisfies all the modality and number constraints on obligatory dattrs and all the necessary structural conditions, then the new mark **nh** is assigned to the individuator with status OWN. After that, it will be subscripted INTERP? and passed across any interpretation cables to nodes in the MRL taxonomy, where it will be stored with status INTERP?. (This amounts to a question of whether the node is a possible interpretation of the construction just built. This is tested by a marker passing strategy which, if successful, results in the new mark **nh** being returned across the interpretation cable with subscript INTERPED and being stored in that same status.) When a node in the syntax taxonomy receives an INTERPED mark that it also OWNs, then that mark is passed up its Superc chains with a subscript that causes the mark to be stored at each node on the chain with status SUP.

5) When a node in the syntax taxonomy receives a SUP mark or an INTERPED mark, it passes it with subscript LMAP to any dattrs of which it is a value restriction or value. Such LMAP marks propagate up chains of **Mods** and **Diffs** links to more general dattrs. Similarly, when a node in the MRL taxonomy receives an INTERP! mark, it passes it with subscript RMAP to any dattrs of which it is a value restriction or value. Again these RMAP marks propagate up Mods and Diffs links to more general dattrs. In addition to this, however, LMAP and RMAP marks will propagate back source and destination links to imap nodes. When an imap

node receives the same mark in both LMAP and RMAP status, it
propagates that mark with subscript IMAP back to its source
dattr. That dattr node, in turn will determine the construction
node that is paired with that constituent node in status
BY-FILLING, and will pass that construction mark with subscript
MAPPED to the node of which this node is a dattr.

6) When a node receives a MAPPED mark, it looks for the
mark pair with status MADE-FROM with that mark as its first
element, and if the second mark of that pair has either ROOT
status or INTERPED status, then the first element is given
INTERPED status. If this results in a mark with OWN status being
given INTERPED status, then this node has a successful
interpretation, and the OWN mark will then be propagated up the
superc chains from the node with subscript SUP, and from this
node and all such Superc nodes it is propagated back along value
restriction links with subscript LMAP to dattrs which can use
this constituent. It is also passed with subscript INTERP!
across the interpretation cable whose imaps enabled it to be
interpreted.

The above algorithm results in an INTERPED mark being
attached to the virtual copy of any constituent in the syntax
taxonomy that has a consistent semantic interpretation in terms
of the semantic interpretation rules that are expressed as
interpretation cables between the syntax taxonomy and the MRL
taxonomy. It also results in a virtual concept corresponding to

the interpretation of the whole input sentence. The algorithm is still somewhat sketchy and remains to be worked out in full detail. However, it is presented here to illustrate the kind of marker passing algorithms that are envisaged, and the way that they work. Many similar algorithms are needed for performing such tasks as verifying the consistency of structural descriptions and value restrictions of virtual concepts, finding the closest matching real node to a virtual concept, finding the most general subsumed concept or the most specific subsuming concept to a given virtual concept, etc.

References

Brachman, R.J. (1977)
"A Structural Paradigm for Representing Knowledge," Ph.D. dissertation, Division of Engineering and Applied Physics, Harvard University.

Fahlman, S.E. (1977)
"A System for Representing and Using Real-World Knowledge," Ph.D. dissertation, Dept. of Electrical Engineering and Computer Science, M.I.T., Cambridge, MA.

Moore, J. and Newell, A. (1973)
"How Can Merlin Understand?" In **Knowledge and Cognition**, L. Gregg (ed.). Maryland: Lawrence Erlbaum Associates.

Quillian, M.R. (1968)
"Semantic Memory," in **Semantic Information Processing** (M. Minsky, ed.). Cambridge, MA: MIT Press, pp. 27-70.

Smith, B. (1978)
"Levels, Layers, and Planes: The Framework of a System of Knowledge Representation Semantics," Master's thesis, M.I.T. Artificial Intelligence Laboratory, February.

Winograd, T. (1975)
"Frame Representations and the Declarative-Procedural Controversy." In D. Bobrow and A. Collins (eds.), **Representation and Understanding**. New York: Academic Press.

Winston, P.H. (1970)
"Learning Structural Descriptions from Examples." Project MAC TR-76, M.I.T., Cambridge, MA.

Woods, W.A. and Brachman, R.J. (1978)
"Research in Natural Language Understanding," Quarterly Technical Progress Report No. 1, Report No. 3742, Bolt Beranek and Newman Inc., Cambridge, MA.

Official Distribution List
Contract N00014-77-C-0378

	Copies
Defense Documentation Center Cameron Station Alexandria, VA 22314	12
Office of Naval Research Information Systems Program Code 437 Arlington, VA 22217	2
Office of Naval Research Code 715LD Arlington, VA 22217	6
Office of Naval Research Code 200 Arlington, VA 22217	1
Office of Naval Research Code 455 Arlington, VA 22217	1
Office of Naval Research Code 458 Arlington, VA 22217	1
Office of Naval Research Branch Office, Boston 495 Summer Street Boston, MA 02210	1
Office of Naval Research Branch Office, Chicago 536 South Clark Street Chicago, IL 60605	1
Office of Naval Research Branch Office, Pasadena 1030 East Green Street Pasadena, CA 91106	1
Office of Naval Research New York Area Office 715 Broadway - 5th Floor New York, NY 10003	1

Naval Research Laboratory   6
Technical Information Division
Code 2627
Washington, D.C. 20380

Naval Ocean Systems Center   1
Advanced Software Technology Division
Code 5200
San Diego, CA 92152

Dr. A.L. Slafkosky   1
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D.C. 20380

Mr. E.H. Gleissner   1
Naval Ship Research & Development Center.
Computation & Mathematics Dept.
Bethesda, MD 20084

Captain Grace M. Hopper   1
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D.C. 20350

Mr. Kin B. Thompson   1
NAVDAC 33
Washington Navy Yard
Washington, D.C. 20374

Advanced Research Projects Agency   1
Information Processing Techniques
1400 Wilson Boulevard
Arlington, VA 22209